DOCUMENT RESUME

ED 264 826 IR 011 909

AUTHOR Bond, Sandra J.; And Others
TITLE Evaluating Computer Manuals for Memorability. CDC Technical Report No. 1.
INSTITUTION Carnegie-Mellon Univ., Pittsburgh, PA. Communications Design Center.
PUB DATE Jan 84
NOTE 39p.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC02 Plus, Postage.
DESCRIPTORS Computer Literacy; *Computers; *Computer Software; *Guides; *Instructional Design; *Instructional Material Evaluation; *Learning Strategies; Long Term Memory; Mnemonics; Psychological Studies; Short Term Memory; Teaching Methods
IDENTIFIERS *Instructional Effectiveness; Text Editing

ABSTRACT

Concerned with the writing and designing of computer manuals that enable users to remember the information presented within them, this study addresses two key questions: (1) whether people remember material longer if they are tested on it immediately after they learn it; and (2) what material they remember (and forget) after a week or month without practice. The study proceeded in three sessions; in the first session, novice computer users were trained on a section of a computer manual and then were tested immediately after the training. The second and third sessions re-tested the subjects 1 week and 1 month later, respectively. Results indicate that people best remember commands with associated mnemonics, commands discussed in detail, and commands presented first or last, supporting the primacy/regency notion of cognitive psychology. Conclusions provide five recommendations for document writers and two recommendations for document and computer designers to increase user retention of material. Five appendices include instructors' comments, excerpts from the manual, and three computer use tests. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED264820

CDC Technical Report No. 1

Evaluating Computer Manuals for Memorability

Sandra J. Bond, Carol J. Janik, and John R. Hayes

January 1984

U.S. DEPARTMENT OF EDUCATION
OERI
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Nicolette Canterna

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The Communications Design Center was established in 1979 to solve communication problems in business, industry, government, and the professions.

The goals of the Center are:

- to encourage interdisciplinary approaches to the solution of visual and verbal communication problems;
- to promote cooperation between communication researchers at the University and those outside the University who have practical interests in communication problems;
- to provide resources and an organizational setting for research in visual and verbal communication;
- to train professional writers, researchers, and teachers;
- to promote the development of educational materials for communication programs on and off the CMU campus.

This report is one of a series from our on-going research on communication and document design. If you have questions or comments on this report, or if you would like to find out more about the activities of the CDC, contact

Communications Design Center
Baker Hall 160
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 578-2906

Abstract

This study looks at the problem of how to write and design computer manuals so that people can remember the information presented in a manual. We were interested in two questions: Do people remember material longer if they are tested on it immediately after they learn it? What material do people remember (and forget) after a week or month without practice? The study proceeded in three sessions. In the first session novice subjects were trained on a section of a computer manual, and then were tested on the information immediately after the training. In the second session one week later the subjects were re-tested. The subjects were also tested one month after the initial training session. Results of the study suggest a number of ways that writers, document designers, and system developers can improve the memorability of computer manuals.

The rapid growth of the computer industry in this country has created a demand for computer documentation that can help the non-expert acquire basic computing skills. Ideally, computer documentation for the novice and occasional user should meet three criteria:

1. It should provide clearly written tutorials to show users how to do something (such as access a system, create a file, etc.).
2. It should help users remember what they have learned.
3. It should allow users to generalize what they have learned to new situations.

The Communications Design Center (CDC) at Carnegie-Mellon University has addressed the first problem with the development of Protocol-Aided Revision, an evaluation tool that helps writers revise a text based on readers' responses to that text. The purpose of the present study is to address the second problem, evaluating documentation for memorability.

In order to examine the memory aspect of documentation, we needed an application that subjects could learn relatively quickly, that was complex enough to provide the data we required, and that we could test reliably. We chose the EMACS text-editing program because it is relatively complex and has a well-defined set of basic commands that novices can learn without having previous computing experience.

Our study proceeded in three sessions. In the first session we trained novice subjects on the EMACS tutorial from the TOPS-20 manual and then tested them on the commands they had learned from the tutorial. In the second session one week later we re-tested the subjects. We also re-tested the subjects one month after the initial session.

We were interested in two questions:

1. Do people remember material longer if they are tested on it immediately after they learn it? and
2. What material do people remember (and forget) after a week or month without practice?

We expected that the subjects would remember the commands with clear mnemonics (such as CTRL d to delete a character) and the commands with only one part, such as ESC d to delete a word, versus commands with multiple parts, such as CTRL @, CTRL w, CTRL y to move a chunk of text. We further expected that subjects would have difficulty remembering inconsistent commands (such as CTRL v to move forward a screen vs. ESC v to move back a screen) and commands with unusual mnemonics (such as CTRL a for "alpha").

Method

Subjects

Using the subject pool in the Department of Psychology, we recruited 10 undergraduates (7 male; 3 female) who had never used EMACS. We chose these subjects because they represented the typical new user of the TOPS-20 manual. The subjects were either paid $3 or were given course credit for their participation in the experiment.

Materials

We developed a standard set of instructions which we read to each subject at the beginning of the training session and before each testing session (see Appendix A).

The training material for this study was a modified version of the EMACS tutorial from the TOPS-20 manual, second edition (see Appendix B). The tutorial gave instructions for using 28 EMACS commands, including how to get into EMACS, how to move the cursor, how to make changes in a text, how to exit EMACS, and how to recover from mistakes.

We also created a practice text file that each subject used to learn the EMACS commands presented in the tutorial (see Appendix C). We used the practice file so that the subjects would not have to take time to type in a chunk of text.

In order to find out what the subjects remembered after the initial training session, we developed two parallel tests with a total of 26 questions (see Appendix D).[1] Form A addressed half the EMACS commands; Form B, the other half. Test items were randomly assigned to both tests. The questions asked how to do something in EMACS and required subjects to recall a particular EMACS command.

In order to examine the mnemonics that subjects used to remember the commands, we also used a modified version of the test questions (see Appendix E). These questions stated the EMACS command and subjects had to recall what the command did.

Procedure

Each subject was trained and then tested individually in three separate sessions.

---

[1] The command for inserting blank lines was not tested, and there were two commands for deleting a character. This explains the difference between the number of test questions and the number of commands presented in the tutorial.

Session 1

At the beginning of the first session we asked subjects to read the manual out loud and to say out loud what they were doing and anything they were thinking. The protocols were tape recorded and subsequently transcribed. The subjects were allowed to practice the commands as much as they felt necessary. The experimenter stayed with the subject and took notes on the subject's activity, reminded the subject to keep talking, and handled any problems with the equipment. After the training period (which took 20-30 minutes) the subjects were given one of the two parallel tests. Half the subjects received Form A; half, Form B. The subjects were allowed to use the terminal and practice file to help answer the questions, but they were not allowed to refer to the manual.

Test Results from Session 1

Even though all 10 subjects were tested immediately after learning EMACS, not one person answered all the test items correctly. This suggests that perhaps we attempted to teach too much too quickly, resulting in cognitive overload. The EMACS tutorial presented 22 single key stroke commands and 6 multiple key stroke commands. Perhaps it is too much to expect a new user to learn that much in one session and to retain it in long-term memory.

Session 2

The second session took place one week later. The subjects were given Form A and Form B, first the one they had not received in the first session, followed by the same test they had received in the first session. The subjects were asked to say out loud everything they were doing and thinking. Again, the protocols were tape recorded and transcribed later. The subjects were allowed to use the terminal and practice file to help answer the questions, but they were not allowed to use the manual.

Test Results from Session 2

The results for the first test of the second session are presented in Figure 1. There were no striking differences between the two tests.

On the average, subjects remembered only 50% of what they had learned one week earlier. One subject, however, answered all questions correctly on the test he had not taken previously. Analysis of his protocol revealed that he was an active learner, explored on his own more than other subjects, tried to make sense of what he was learning, and had a positive attitude toward learning the material. All these factors may have contributed to his retention of material one week after he'd learned it. Here is an excerpt from his protocol:

$\bar{X}$ # Errors	Question	Command
0	right a space	CTRL f
0	left a space	CTRL b
0	end of line	CTRL e
0	end of sentence	ESC e
.17	delete a character	CTRL d
.17	down a line	CTRL n
.17	left a word	ESC b
.25	right a word	ESC f
.25	beginning of line	CTRL a
.25	reformat screen	CTRL l
.25	back one screen	ESC v
.25 (2 parts)	enter EMACS	EMACS file.name
.33	cancel previous command	CTRL g
.33 (2 parts)	recover after CTRL c	continue <CR>
.5	delete rest of line	CTRL k
.5	forward one screen	CTRL v
.5	beginning of file	ESC <
.5	delete a word	ESC d
.5	up a line	CTRL p
.5	insert characters	type them in
.5	end of file	ESC >
.5	beginning of sentence	ESC a
.5 (multiple parts)	move a chunk	CTRL @ CTRL w CTRL y
.67 (multiple parts)	save	CTRL xs
.75 (multiple parts)	delete a chunk	CTRL @ CTRL w
.79 (multiple parts)	leave EMACS	CTRL xs CTRL xz

Figure 1. Results of first test one week after learning EMACS

Moving the cursor left or right to the beginning or end of a line. The cursor left or right to the beginning or end of a line. Action 1. Alternately do CTRL e (end) and CTRL a (alpha)... Response. The cursor will move from the beginning to the end of the line, or vice versa. No kidding. Practice using these six commands to move the cursor to various positions in the file. You may want to use the Summary on page C-16. I don't even know where C-16 is--they ought to be page numbered. Oh, well, I don't need to use the summary anyways, but--that's a piece of document change you have to do here--put page numbers on things. See what we're doing here... "e" to end line... See what happens when "a"--do "x"--What was that? Oh, that was just "n"...I think I've edited it inadvertently here. It stays at the end of the line--ok... It goes up to--I'm curious about that. Let's see... It stays the same distance, or roughly the same distance. Yeah, it does. Until you get to the end of the line...Well...Ok. Great fun. Learned a few things about that.

Session 3

We called subjects back one month later without their knowing we would, and we tested them again. This time, though, we did not tape record the session, and we gave them the modified test instead. For example, instead of asking, "How do you move to the end of a line?" we asked, "What's Control e? What does the e stand for?" We wanted to see if people could remember commands that had mnemonics associated with them better than those that did not.

Test Results from Session 3

Results suggest that commands without associated mnemonics are remembered about as well as commands with associated mnemonics. This result, however, is confounded, since at least half the subjects had been using EMACS regularly since they had learned it.

Only one subject had not used EMACS at all since the initial session, and his test score reflected this. He made more errors than any other subject, due probably to simple forgetting and lack of practice. Of the eight commands he remembered correctly, three had an associated mnemonic in the EMACS tutorial, but five did not. (The five commands were: ESC >, ESC <, CTRL g, CTRL L, and CTRL xs CTRL xz.) For two of those five nonmnemonic commands, though, he had created his own mnemonics. Two of the other nonmnemonic commands he remembered (Control g and Control L) were discussed in more detail in the tutorial than other commands. The final nonmnemonic command he remembered--leaving EMACS (Control xs Control xz) was the last item presented in the tutorial. It appears, then, that after a month, people best remember commands with associated mnemonics, commands discussed in detail, and commands presented first (two of his three correct mnemonic commands were the first ones presented in the tutorial), or last, supporting the primacy/recency notion of cognitive psychology.

Protocol Results

In examining the protocols from Sessions 1 and 2, we looked at three categories of errors:

- misremembered commands - (not remembering a command at all or using a command incorrectly)
- confused commands - (using an Escape command instead of a Control command, and vice versa), and
- invented commands - (using a command not learned in the EMACS tutorial)

These categories, of course, are arbitrary. Some errors fell into all three categories. Other "errors" (for example, not being able to assign a name to a key (e.g., @, <, >)) fell into none.

Misremembered Commands

Subjects were allowed to use the terminal while they were being tested in Sessions 1 and 2, and indeed this helped some subjects to recall some commands through experimentation with the EMACS practice file. But some subjects still could not reconstruct what they had learned earlier, for example:

> How do you reformat the screen to get rid of noise or system messages? Let me see. That's probably a CTRL..Let me see. I forget that letter--I'll have to come back to that.

In the example above, the subject forgot the answer, CTRL L. CTRL L is presented in the tutorial without an accompanying mnemonic. Lest we be too hasty in concluding that nonmnemonic commands are harder to remember than those with an associated mnemonic, consider the following example:

> How do you delete a word? Oh, that's worse. Move the cursor..to the--was it the first or the last?..I think it was the last character--last character in the word and hit--what was that?--What was that? I don't remember that one. I think it was ESC something. ESC and a letter.

In this example, the subject could not recall ESC d (for delete), even though the word "delete" was cued into the question.

We believe we must conclude that lack of practice, simple forgetting, and perhaps presenting subjects with too much information in one sitting, resulting in cognitive overload, all contribute to errors of the above type.

Some subjects simply misremembered what they had learned a week earlier. While they

were able to recall specific commands, they had difficulty remembering which commands did what, for example:

> How do you delete the rest of a line?...Maybe if you move the cursor to the end of a line...and press ESC d. I'm not sure what it's doing. Press ESC d again. It's just doing words again. So I'm not sure how you do that.
>
> How do you move left one word? Well CTRL w, I think that was the wrong thing to push since that might be wipeout...

Both ESC d and CTRL w are legitimate EMACS commands presented in the tutorial, but they do not do what the two subjects above tried to make them do. Again, we conclude that subjects were presented with too much information at orice, resulting in confusion in remembering which commands did what.

Some subjects could not remember the mnemonic associated with a command, resulting, of course, in their also not being able to recall the command.

> How do you move up a line? I still don't know that. And I know that's an easy command. It's just the opposite of CTRL n. And I seem to have it in the back of my mind, but I just can't remember it. So CTRL n - what goes with CTRL n to move up?

Other subjects recalled a command but associated the wrong mnemonic with it, resulting in errors. In the examples below, both subjects incorrectly associated CTRL b (for back one space) with the word "beginning."

> CTRL b - beginning of line...
>
> How do you move to the beginning of a sentence?...CTRL b...It's CTRL something. I'm not quite sure.

In the following examples, the two subjects correctly recalled legitimate EMACS commands as well as their associated mnemonics (ESC a = alpha, which means "beginning" in Greek; ESC e = end), but they misremembered what the commands did:

> Beginning of a file? That's ESC a for alpha.
>
> How do you move to the end of a file? It might be ESC e...

ESC a moves the cursor to the beginning of a sentence (not the beginning of the file), and ESC e moves the cursor to the end of a sentence (not the end of the file).

In learning information that is not inherently meaningful to a new user, mnemonics can be extremely useful. But our subjects had difficulty both recalling the mnemonics presented in the EMACS tutorial, as well as associating the correct mnemonics with their appropriate

commands. This is a particularly troublesome problem to solve in EMACS, since EMACS is designed to be an efficient, time-saving editor. While changing commands from a single key stroke to, for example, a truncated, meaningful command (e.g., changing b to beg for beginning or bk for back) might improve accuracy among new and occasional users, it would be frustrating and inefficient for experts. The problem to be solved, then, is to devise unique mnemonics which do not trigger multiple associations (like back and beginning) and which, at the same time, are obvious and meaningful. Half our subjects could not remember CTRL p (for previous), even though the association appears obvious to us. We believe we can improve the EMACS tutorial by reinforcing the mnemonics within the text. Figure 2 shows a portion of the original tutorial material with suggested changes hand-written in. We hypothesize that changes like these will result in fewer errors by EMACS users.

When subjects could not remember the correct answer to a question, they often resorted to random search, as in the following examples:

> Let's move up a line. Try that one again. That's CTRL and some letter. So it's not CTRL q, w, e, r, t, y, g - CTRL f - no, that's forward. CTRL b - CTRL b doesn't do anything. CTRL f - CTRL b...
>
> How do you move back one view screen? It's got to be CTRL something, but I don't know what. CTRL s - CTRL u - CTRL w - CTRL k...I don't know what I'm doing here...
>
> How do you move back one view screen?...OK, OK. I could do--let's see--CTRL xz--or is it CTRL xm? CTRL xm? Or CTRL nx--nz--Well, I got out of it anyway is that the idea?
>
> E: No, it isn't.
>
> OK...CTRL sz--no. CTRL--CTRL mn--CTRL mx...

The above errors, while probably frustrating for the subjects, are fairly innocuous. But a number of subjects who resorted to random search made critical (and sometimes fatal!) errors. Three subjects actually managed to delete the entire file they were working in! Following are examples of critical errors from the subjects' protocols.

> How do you move forward one view screen? It's not CTRL z. CTRL w? I know CTRL w does something. Maybe that's on save your work...I don't know how...OK, I'm going to type CTRL c.

Typing CTRL c is a critical error because it exits the file without saving any work. It is obvious from the example above that this subject was trying random search and simply happened upon CTRL c. The following two examples, though, are more egregious:

2. Moving the cursor ~~left or right~~ forward or back one space.

Action 1. Do a CTRL f (forward) several times.

Response. The cursor moves ~~right~~ forward several spaces.

Action 2. Do a series of CTRL f's until the cursor moves to the extreme right of the screen. Then do several more.

Response. The cursor moves to the ~~beginning of the~~ forward next line.

Action 3. Do a series of CTRL b's (back).

Response. The cursor moves ~~left~~ back until it reaches the beginning of the line. Then, with continual CTRL b's, the cursor moves to the ~~right~~ back end of the previous line.

3. Moving the cursor ~~left or right~~ to the beginning or end of a line.

Action 1. Alternately do CTRL e (end) and CTRL a (alpha).

Response. The cursor will move from the beginning to the end of the line, or vice versa.

Practice using these six commands to move the cursor to various positions in the file. You may want to use the Summary on page C-16 for easy reference.

Figure 2. A section of the EMACS tutorial revised to reinforce mnemonics

> How do you leave EMACS? CTRL c. That's all I know how to do.
>
> How do you cancel a previous EMACS command?...I think you do two ESC commands.

The text from the tutorial which discusses CTRL c and cancelling a previous EMACS command are presented in Figure 3.

Both the above subjects recalled our examples of what not to do as how to do something. As document writers, this presents us with a dilemma. While users obviously need to know how to avoid making critical (and fatal) errors, having them practice the error and then telling them how to recover from it may not be the best strategy. When tested one week after learning EMACS, one-third of our subjects said that the way to leave EMACS was CTRL c!

Frankly, we don't know an optimal solution to this problem. One possibility would be to exaggerate the consequences--e.g., if you type CTRL c, your terminal will explode--although we suspect such measures would not facilitate learning. Another possibility would be to separate out the critical error information from the rest of the text and emphasize it more, perhaps with headings, boldface type, and boxes. We will try these latter measures in further revisions and testing of the EMACS tutorial. Finally, we recommend that the terminal keyboard or an accompanying keyboard snap-on insert or overlay of some type be redesigned as much as possible to integrate critical errors with a warning system. Working with a graphic designer and/or human factors expert could produce optimal results in redesigning keyboard features. At a minimum, though, we recommend that the "c" key be distinguished from the rest of the keyboard. For example, it could be colored red to serve as a warning similar to the red buttons on tape recorders for erase and record.

Six of the 28 EMACS commands our subjects learned involved multiple steps. These commands proved particularly difficult for subjects to remember after a week without practice. Of the six multiple step commands, over half the subjects missed four of them, and what is more, they missed them completely. With rare exception, subjects either remembered all of the steps or none of them, as the following example shows:

> How do you delete a chunk of text? You move the cursor to the beginning and hit--I forget the keys--then you move the cursor to the end of the material--you need another combination of keys which I also forget.

Particularly distressing for us and frustrating for the subjects is that after one week not one subject could correctly recall how to leave EMACS. The correct command (CTRL xs, CTRL xz) may be hard to remember because it requires multiple steps and has no

Emergency Procedures

CTRL G is the ultimate panic button. It is useful when you have made a mistake in using an EMACS command. For example, suppose that you want to move ahead one view screen and you mistakenly press ESC rather than CTRL. You can cancel the "escape" by doing a CTRL g before you issue another command.

Action 1. Hit ESC.

Response. The message M- appears in the lower left corner of the screen, followed by the cursor. The M indicates that the ESC (or Meta) key has been pressed. The dash followed by the cursor indicates that EMACS is waiting for you to complete the command by typing a letter such as "b" or "v".

Action 2. Do a CTRL g

Response. The message M-^G appears at the lower left part of the screen and the cursor returns to the position it occupied before you pressed the ESC key.

Another common problem occurs if you accidentally press the ESC key twice. If you do so, you will be put into "Minibuffer," an advanced feature of EMACS which we won't discuss further in this tutorial. For the novice, Minibuffer is trouble. Doing one or two CTRL gs will get you out of Minibuffer.

Action 1. Hit ESC twice.

Response. A row of dashes appears a few lines from the top of the screen and "(Minibuffer)" appears in the lower left.

Action 2. Do a CTRL g

Response. Your file reappears.

Figure 3. An excerpt from the EMACS tutorial explaining CTRL c and CTRL g

CTRL L instructs EMACS to blank and then re-draw the screen. It is especially useful if your screen has been "damaged" in some way--for example, by a system message or by noise (characters that appear on your screen due to interference on the communication line). Since EMACS' efficiency depends upon the fact that you and EMACS both know at what character-position in the file you are working, it is vital that you agree on that point. CTRL L will get rid of noise and will ensure that you and EMACS are looking at the same position in the file indicated by the cursor. In addition, CTRL L may reconfigure the screen so that the present cursor position, if it was near the bottom or top of the screen, is at the center of the screen.

You may want to do a CTRL L to see how it works.

CTRL C returns you the Exec. It is *not* the right way to leave EMACS. If you strike it by mistake (easy to do, since there are so many useful CTRL-something commands) you can be put right back where you and EMACS were by issuing the simple Exec command *continue*

Action 1. Do a CTRL c

Response. The @ prompt appears.

Action 2. Type *continue* and hit RETURN; then do a CTRL L

Response. The screen appears as it did before you did CTRL c.

If you would like to continue practicing using EMACS, you may use the Summary on the following pages, or you may go back to the tutorial for more explicit instructions. If not, you should leave EMACS now by doing CTRL xs CTRL xz (or if you have a save command in your Emacs.Vars file, just CTRL xz).

Postscript:

We have covered only the most basic aspects of EMACS in this tutorial. If you would like to learn more about EMACS, you can obtain an on-line tutorial by typing the Exec command *emacs:teach-emacs* and hitting RETURN.

Figure 3. (continued)

associated mnemonics in the tutorial. Subjects who do remember commands with multiple steps may be using a chunking strategy or a personal mnemonic not presented in the text. One subject created his own mnemonic for the second part of the leave EMACS command:

> How do you leave EMACS? Well, I know how to do that actually. CTRL xz - for exit.

This subject got this part of the command right after one week and also after one month.

It is important to remember that the tutorial we used in this experiment was an abbreviated version of the TOPS-20 EMACS tutorial. This abbreviated version did not discuss saving work and leaving EMACS in nearly the detail the original did. Perhaps a more elaborated and detailed explanation of multiple step commands would have aided subjects' recall. However, we think the optimal solution overall would be to simplify EMACS commands with multiple steps, especially the command for leaving EMACS, since it is such an important command. Short of that, perhaps the best heuristic is to generate unique mnemonics when they are lacking, even if the mnemonics appear forced. Some mnemonic is often better than no mnemonic.

Confusion between Commands

Using a control command when an escape command was appropriate (and vice versa), or not knowing which one to use, was a common problem across subjects, as the following examples show:

> Forward one view screen? I think it's ESC v. One moves you forward, the other moves you back. CTRL v - ESC v. So I'll go with ESC v for forward.

> How do you move back one view screen?...One view screen, that was CTRL--one view screen--was that CTRL or ESC? I'm not quite sure.

> How do you move forward one view screen? That was a "v" but was it CTRL or--

> OK, CTRL v moves it back a screen and ESC v moves it forward a screen.

> Forward one view screen. Well, I'm pretty sure it's either ESC or CTRL, but I don't know which one...

In the tutorial, we discussed control commands before we discussed escape commands. Control commands are simpler and more local than escape commands and we wanted to build our subjects up from learning the basic commands to learning the meta commands. When we finished discussing control commands, this is how we introduced escape commands:

> Additional move commands are available using the ESC key; these commands are sometimes called "Meta" commands. Commands involving the ESC key are somewhat "smarter" than the CTRL key commands. That is, meta commands know how to recognize words and sentences.

We thought our presentation of escape commands was clear and sufficient, but when we tested our subjects, we discovered that they remembered the control (local) commands much better than the escape (meta) commands. This resulted in gross inefficiencies like the following:

> How do you delete a word? Let me see. Move the cursor to the first character in the word, then press CTRL d - as many characters as are in the word. There's probably an easier way to do it but I forget.
>
> How do you move to the end of a line? For this question, do they want like one--how do you do it in one--in one pressing? Because I can move to the end of the line using several CTRL f's.
>
> How do you move left one word? You can use several CTRL b's...
>
> How do you delete the rest of a line?...You can use several CTRL d's and that will do it.
>
> How do you move right one word? I know that you can--one method of doing it is using several CTRL f's...
>
> How do you delete a word?...All right, you can use several CTRL d's...There's another way of doing it, too, but I don't remember.

Because meta commands are some of the most efficient EMACS commands, we believe it is important that new users be able to remember them. We were distressed that two subjects who returned for testing after one month told us they never use meta commands, only control commands: "Even though they're less efficient, they work." In future revisions of the tutorial we will change the presentation of meta commands. In one version, we will present the escape commands first to test the primacy theory. In another version, we will present corresponding control and escape commands together, e.g., CTRL f = forward one character; ESC f = forward one word. Hopefully one or both of these changes in presentation will result in better memory for meta commands.

Invented Commands

The most intriguing errors were what we call "invented commands," that is, commands that subjects generated when we tested them but which had not been presented in the EMACS tutorial. Of course, some of these commands simply resulted from random search. For example:

> How do you move back one view screen? CTRL t - next to nothing. I just did ESC t. I figured I hadn't been using t's, but at least see what it does.

> How do you move back one view screen? It's got to be CTRL something, but I don't know what. CTRL s--CTRL u--CTRL w--CTRL k...I don't know what I'm doing here...

But some of the invented commands did not appear to be random. Indeed subjects seemed to be using them quite deliberately:

> How to delete a word. I know, I know, CTRL q.

The following comment was the most intriguing:

> How do you move a chunk of text? OK, you move to the beginning of what you want to move. So--let's see. Maybe this sentence...You do a--you don't do a CTRL b. I know you have to do a CTRL w to get it back. CTRL k, maybe. Oh, no. I'm remembering Pascal now.

We had requested subjects with no knowledge of EMACS, but when certain errors--in particular, CTRL q, ESC n, and ESC p--kept showing up in testing across subjects, we began to suspect that some prior knowledge of computers was interfering with our subjects' learning EMACS. We already knew that some of the commands, like CTRL q, were TOPS-20 executive level commands. To solve the enigma of the rest of them, we showed them to a computer expert in the CMU Psychology Department and asked him, "Other than in EMACS, when might one use these commands?"

He told us that many of the commands work in ALOE, a language oriented editor specifically designed to work on Pascal programs. While ALOE is, technically speaking, an editor, unlike EMACS, which works with text files, ALOE works with structures. (A structure in ALOE refers to a boolean (if-then) function.) For example, in ALOE, ESC p moves the cursor out one structure; ESC n, in one structure.

ALOE is currently taught in conjunction with the CORE computer course. Seven of the 10 subjects were freshmen in the CORE and therefore taking the computer course. It is highly conceivable, therefore, that our subjects confused prior knowledge of ALOE with new knowledge of EMACS. This may account for many of the invented commands subjects generated in testing.

Interference from prior knowledge of ALOE may also have contributed to errors in learning EMACS. For example, some commands, e.g., CTRL k, are used in both editors, but have different results. In EMACS, CTRL k kills the line of text to the right of the cursor. In ALOE, CTRL k deletes a structured region in a Pascal program.

Maximum interference seems to occur when the same commands in different systems have somewhat but not completely overlapping results, e.g., both delete something, but not the same thing. The optimal solution for avoiding both proactive and retroactive interference is to construct unique commands for every editor.

Conclusions

Here are our recommendations for document writers:

1. Teach less information in a single session. Since only one subject answered all test questions correctly one week after learning EMACS, it appears that we taught too much information (28 commands) in a single session.

2. Encourage practice, especially for the occasional user. A week's delay in learning EMACS and using it again is simply too long for people to remember the information they learned initially.

3. In the tutorial, use unique mnemonics and reinforce them verbally (and, when possible, visually) in the text.

4. Avoid using common words in the text which are also commands, e.g., several CTRL b's move you to the beginning of a line.

5. Present the most important commands first to take advantage of the primacy effect. Examples would be entering EMACS, leaving EMACS, and ESC commands.

Here are our recommendations for document and computer designers:

1. Redesign the keyboard or provide a keyboard insert or overlay designating keys which, when pressed, could have critical results. For example, in EMACS, the "c" key could be color-coded red.

2. Design unique commands to avoid interference from prior knowledge of other systems.

Learning EMACS

In this experiment, you will learn to use EMACS, the TOPS-20 editing system. I will give you part of a manual explaining how to use EMACS. I want you to work through the manual, saying everything out loud as you do so. That is, I want you to read out loud. When you use the terminal, say out loud what it is you are doing, and anything you are thinking, no matter how trivial it may seem to you. I am here to tape record what you say, to take notes on what you are doing, to remind you to keep talking, and to make sure the equipment works right. Try to pretend that I am not here and that you are working through the manual on your own. If you have any questions, please ask them before you start reading the manual.

This is a two-part study. Part 2 will be very similar to Part 1. It is very critical to the success of this experiment that you NOT practice what you learn today in Part 1 before you come back to do Part 2. Will you agree NOT to practice?

To work in EMACS, you need to use two non-standard typewriter keys. One is the Control key, and the other is the Escape key. [Show subjects the keys.] You use the Control key in conjunction with another keyboard character. To use the Control key, hold the key down and simultaneously strike the keyboard character you want. Then release both keys. To use the Escape key, press and release the Escape key, and then proceed with the instructions associated with the Escape key.

Do you have any questions before beginning? [answer questions]

I'm going to turn on the tape recorder now. When I hand you the manual, you may begin learning EMACS. Remember--say everything out loud.

Now that you have completed Part 1 of Learning EMACS, I'd like to give you a brief quiz to see what you remember. Again, please say everything out loud as you take the quiz. You can use the terminal and practice file to answer the questions, but please also record your answers on the sheet of paper I will give you. You cannot, however, use the manual.

Debriefing Statement

You have just participated in a memory study. In Session 1, you learned how to use EMACS, the TOPS-20 editing system. In Session 2, we tested you to see what you remembered, as well as what you had forgotten, about EMACS after a week's delay. We did not tell you that we were going to test you because we wanted to simulate the natural way in which infrequent users learn to use the computer. That is, if we had told you you would be tested, you might have studied the material in a different, non-natural way.

We are interested in those aspects of learning to use a computer that are both easy and difficult to learn and remember, especially for the infrequent user who typifies many faculty and staff members at CMU. Our goal is to make the computer easy to learn and use, even for the infrequent user, by improving computer software and documentation. The information you have given us will help us move closer to that goal. Thank you.

EMACS is an easy to use editing program which allows you to modify a file that already exists by adding material, deleting material, or moving material from one place in the file to another.

To edit with EMACS, first you must enter EMACS and designate the file you want to edit.

Action: At the @ prompt, type *emacs practice.txt* then hit RETURN.

Response: Your file "practice.txt" (or as much as will fit) appears on the screen and the following message appears at the bottom:

```
EMACS (Text Fill Save) Main:  Practice.txt
```

If your file occupies more than one screen EMACS adds the message --TOP-- after the file name to indicate that you are looking at the first part of the file.

Using EMACS to change a file typically involves two steps:

1. Moving the cursor to the part of the text you want to change by using the EMACS move commands.

   Note: Don't use the spacebar, RETURN, BACKSPACE, or DELETE keys to move the cursor.

2. Making changes, e.g., an insertion, a deletion, etc., by using other EMACS commands.

First we describe the commands for moving the cursor, and then the commands for making changes.

Note: If the system is very busy, there may be a pause between when you give an EMACS command and when you see the results on your screen.

A. Moving the cursor

The commands in this section are condensed for easy reference in the Summary at the end of this tutorial.

1. Moving up or down a line at a time.

Action 1. Do a CTRL n (next).

Response. The cursor moves down one line.

Action 2. Do 4 or 5 CTRL n's.

Response. The cursor moves down 4 or 5 lines.

Action 3. Do a CTRL p (previous).

Response. The cursor moves up a line.

Action 4. Do 5 or 6 CTRL p's.

Response. The cursor should now be at the top of the screen.

Action 5. Do several more CTRL p's.

Response. Notice that CTRL p has no effect when the cursor is already at the first line of the file.

Action 6. Do a series of CTRL n's so that the cursor moves to the bottom of the screen. Now do several more CTRL n's.

Response. The screen scrolls up; that is, the lines of the file move up, some disappearing from the top of the screen. Notice also that the indicator at the lower right which used to say --TOP-- now says something else, such as --32%-- or --BOT-- indicating that you are now part way through the file or at the bottom of the file.

Action 7. Do a series of CTRL p's until the cursor is back at the first line of the file.

Response The screen scrolls down until the beginning of the file is at the top of the screen.

2 Moving the cursor left or right one space.

Action 1. Do a CTRL f (forward) several times.

Response. The cursor moves right several spaces.

Action 2. Do a series of CTRL f's until the cursor moves to the extreme right of the screen. Then do several more.

Response. The cursor moves to the beginning of the next line.

Action 3. Do a series of CTRL b's (back).

Response. The cursor moves left until it reaches the beginning of the line. Then, with continual CTRL b's, the cursor moves to the right end of the previous line.

3 Moving the cursor left or right to the beginning or end of a line.

Action 1. Alternately do CTRL e (end) and CTRL a (alpha).

Response. The cursor will move from the beginning to the end of the line, or vice versa.

Practice using these six commands to move the cursor to various positions in the file. You may want to use the Summary on page C-16 for easy reference.

Additional move commands are available using the ESC key; these commands are sometimes called "Meta" commands. Commands involving the ESC key are somewhat "smarter" than the CTRL key commands. That is, meta commands know how to recognize words and sentences. In particular, EMACS defines a sentence as any group of characters followed by a punctuation mark and two spaces.

Note: The procedure for using the ESC key is slightly different from that for using the CTRL key. Rather than holding the ESC key down, strike it once and then strike the letter.

4. Moving the cursor right or left one word.

Action 1. Strike ESC and then strike f (forward).

Response. The cursor moves to the beginning of the next word.

Action 2. Alternately strike ESC and f several more times.

Response. The cursor moves several words to the right.

Action 3. Strike ESC and then strike b (back).

Response. The cursor moves to the beginning of the previous word.

5. Moving the cursor to the beginning or end within a sentence.

Action 1. Do an ESC e (end).

Response. The cursor moves to the end of the sentence in which it was located. In particular, it moves to the space after the period.

Action 2. Do an ESC e again.

Response. The cursor moves to the end of the next sentence.

Action 3. Do an ESC a (alpha).

Response. The cursor moves to the first letter of the sentence in which it was located.

Action 4. Do another ESC a

Response. The cursor moves to the first letter of the previous sentence.

6. Moving the cursor forward or back one screen page

Action 1. Do a CTRL v (view).

Response. The current screen page is replaced by the next screen page of the file.

Action 2. Do an ESC v (view).

Response. The current screen page is replaced by the previous screen page of the file.

7. Moving the cursor to the beginning or end of the file.

Action 1. Do an ESC > That is, strike the ESC key, hold down the SHIFT key and strike >

Response. The cursor moves to the end of the file.

Action 2. Do an ESC <

Response. The cursor moves to the beginning of the file.

B. Making changes

The commands in this section are also condensed for easy reference in the Summary at the end of this tutorial.

1. Inserting

A. To insert new characters

Action. Use the appropriate EMACS move commands to position the cursor where you want the new characters to appear and type them in.

Response. The characters you type are inserted immediately to the left of the cursor.

B. To insert a blank line (for example, between paragraphs or items in a list)

Action. Position the cursor at the end of the paragraph or list and hit RETURN *twice*.

Response. A blank line will be inserted just before the cursor.

2. Deleting

A. To delete a character

Action. Position the cursor after the character you want to delete and hit the DELETE key.

OR, Position the cursor on the character you want to delete and do a CTRL d (delete).

Response. The character disappears.

B. To delete a word

Action. If you are going to replace the word, position the cursor on the first letter of the word and do ESC d (delete).

OR, If you are not going to replace the word, position the cursor on the space before the word and do ESC d (delete).

Response. The word disappears.

C. To delete the rest of a line

Action. Position the cursor where you want the deletion to start and do a CTRL k (kill).

Response. The line becomes blank from the cursor to the right margin.

D. To delete a chunk of material

Action. Position the cursor where you want the deletion to start.

On a Zenith, Fox, or VT52 terminal, do a CTRL @ (On some terminals this requires you to hold down both the CTRL and the SHIFT key while you strike the @ key.)

On a VT100 terminal, do a CTRL space

Then, using the other move commands, position the cursor immediately after the last character you want to delete and do a CTRL w (wipeout).

Response. The chunk of material disappears.

3. Moving a chunk of material

Action 1. First you must delete the chunk of material you want moved. Follow the procedure above to delete a chunk of material--don't be alarmed when it disappears.

Action 2. Then, using the other move commands, position the cursor at the point where you want the chunk inserted and do a CTRL y

Note: Do not use any of the EMACS delete commands while you are positioning the cursor.

Response. The chunk of material reappears at the point where you typed the CTRL y

Note: CTRL y always acts on the previous CTRL @/CTRL w or CTRL space/CTRL w If you make a mistake using these commands, do a CTRL g (explained in the following section) before doing a CTRL y

Emergency Procedures

CTRL G is the ultimate panic button. It is useful when you have made a mistake in using an EMACS command. For example, suppose that you want to move ahead one view screen and you mistakenly press ESC rather than CTRL. You can cancel the "escape" by doing a CTRL g before you issue another command.

Action 1. Hit ESC.

Response. The message M- appears in the lower left corner of the screen, followed by the cursor. The M indicates that the ESC (or Meta) key has been pressed. The dash followed by the cursor indicates that EMACS is waiting for you to complete the command by typing a letter such as "b" or "v".

Action 2. Do a CTRL g

Response. The message M-^G appears at the lower left part of the screen and the cursor returns to the position it occupied before you pressed the ESC key.

Another common problem occurs if you accidentally press the ESC key twice. If you do so, you will be put into "Minibuffer," an advanced feature of EMACS which we won't discuss further in this tutorial. For the novice, Minibuffer is trouble. Doing one or two CTRL g's will get you out of Minibuffer.

Action 1. Hit ESC twice.

Response. A row of dashes appears a few lines from the top of the screen and "(Minibuffer)" appears in the lower left.

Action 2. Do a CTRL g

Response. Your file reappears.

CTRL L instructs EMACS to blank and then re-draw the screen. It is especially useful if your screen has been "damaged" in some way--for example, by a system message or by noise (characters that appear on your screen due to interference on the communication line). Since EMACS' efficiency depends upon the fact that you and EMACS both know at what character-position in the file you are working, it is vital that you agree on that point. CTRL L will get rid of noise and will ensure that you and EMACS are looking at the same position in the file indicated by the cursor. In addition, CTRL L may reconfigure the screen so that the present cursor position, if it was near the bottom or top of the screen, is at the center of the screen.

You may want to do a CTRL L to see how it works.

CTRL C returns you the Exec. It is *not* the right way to leave EMACS. If you strike it by mistake (easy to do, since there are so many useful CTRL-something commands) you can be put right back where you and EMACS were by issuing the simple Exec command *continue*

Action 1 Do a CTRL c

Response The @ prompt appears.

Action 2 Type *continue* and hit RETURN; then do a CTRL L

Response The screen appears as it did before you did CTRL c.

If you would like to continue practicing using EMACS, you may use the Summary on the following pages, or you may go back to the tutorial for more explicit instructions. If not, you should leave EMACS now by doing CTRL xS CTRL xz (or if you have a save command in your Emacs.Vars file, just CTRL xz).

Postscript:

We have covered only the most basic aspects of EMACS in this tutorial. If you would like to learn more about EMACS, you can obtain an on-line tutorial by typing the Exec command *emacs:teach-emacs* and hitting RETURN.

EMACS SUMMARY

To get into EMACS	Type *emacs <name of file>*
To help save your work	Do CTRL xs
To leave EMACS	Do CTRL xs CTRL xz If you have a save command in your Emacs Vars file, just do CTRL xz

Moving the Cursor

	Forward	Back
1. Up or down a line	CTRL n (next)	CTRL p (previous)
2. Left or right a space	CTRL f (forward)	CTRL b (back)
3. To the beginning or end of a line	CTRL e (end)	CTRL a (alpha)
4. Left or right one word	ESC f (forward)	ESC b (back)
5. To the beginning or end of a sentence	ESC e (end)	ESC a (alpha)
6. Forward or back one view screen	CTRL v (view)	ESC v (view)
7. To the beginning or end of the file	ESC >	ESC <

Making Changes

1. Inserting new characters	Position the cursor where you want the new characters and type them in.
2. Deleting a character	Position the cursor after the character and hit DELETE OR, position the cursor on the character and do a CTRL d
3. Deleting a word	Position the cursor on the first letter and do an ESC d OR, position the cursor on the space before the first letter and do an ESC d
4. Deleting rest of line	Position the cursor where you want the deletion to start and do a CTRL k
5. Deleting a chunk of text	Position the cursor where you want the deletion to start. On a Zenith, Fox, or VT52 terminal, do a CTRL @ On a VT100 terminal, do a CTRL space Then position the cursor after the last character you want to delete and do a CTRL w
6. Moving a chunk of text	Follow the procedure for deleting a chunk. Then position the cursor where the chunk should be inserted and do a CTRL y

Emergency Procedures

CTRL g	cancels the previous EMACS command
CTRL L	reformats the screen and gets rid of noise or system messages
continue	type this at the Exec to get back into EMACS if you did a CTRL c by mistake

Mars is, next to the Moon, our closest neighbor in the solar system. It is a small planet, about half as big in diameter (4,200 miles) as the Earth, and with only one tenth of the Earth's mass. Its day is about the same as ours (a little more than 24 1/2 hours), and its year--the period of revolution around the Sun--is 687 days. Its four seasons, unlike the Earth's, vary considerably in length.

Mars has a decidedly eccentric orbit, and its nearest approach to us comes at intervals of about 15 years: the next one will occur in 1986. At its closest it is about 35 million miles from us--150 times farther away than the Moon. Even with our largest telescopes we can get no better view of Mars than we can of the Moon through low-power binoculars. Nevertheless, the long, patient visual examination of the planet by a few astronomers who took a special interest in Mars--notably the Italian Giovanni Schiaparelli, the American Percival Lowell, and the Frenchman Eugene Antoniadi--and recent physical studies have given us a fairly detailed description of the main features of the planet's surface and atmosphere. Today the study of Mars has achieved the status of a full-fledged branch of astronomy.

Easily the most conspicuous feature of the planet is the white caps that cover its polar regions. They display a fascinating rhythm of advance and retreat. At the end of winter in each hemisphere the polar cap covers some four million square miles. As spring comes, it begins to diminish--rather slowly at first, then at an increasing rate. Near the middle of spring dark rifts appear. They grow steadily and soon split the cap into several sections. Disintegration of the fragments then proceeds rapidly. But the cap never disappears completely: even in midsummer a tiny dazzling spot remains near the pole.

As summer draws to its end, diffuse patches of whitish cloud appear in the polar region. They spread quickly. With the arrival of fall they drop a curtain over the entire scene, covering not only the polar area but part of the "temperate" belt. The rather bright and unsteady veil persists through the fall and winter, concealing from our view the icecap re-forming beneath it. By the end of winter the cloud cover breaks up and the polar cap reappears, somewhat dim at first, then growing bright.

Year after year this cycle follows the same general pattern. But its regularity is far from perfect: the cap may vary slightly in size from year to year. Some astronomers have suggested that the irregularities may be correlated with the 11-year sunspot cycle, but we have no definite evidence of this.

Appendix D

Name _____

Date _____

EMACS Quiz (Form A)

1. How do you get into EMACS?

2. How do you reformat the screen to get rid of noise or system messages?

3. How do you move right one word?

4. How do you delete a word?

5. How do you move up a line?

6. How do you insert new characters?

7. How do you move to the beginning of a line?

8. How do you move to the end of the file?

9. How do you move right a space?

10. How do you delete a chunk of text?

11. How do you move back one view screen?

12. How do you move to the beginning of a sentence?

Name _____

Date _____

EMACS Quiz (Form B)

1. How do you move to the end of a line?

2. How do you move a chunk of text?

3. How do you save your work?

4. How do you delete a character?

5. How do you move down a line?

6. How do you cancel the previous EMACS command?

7. How do you move left one word?

8. What can you do to recover if you accidentally type a CTRL c?

9. How do you move left a space?

10. How do you delete the rest of a line?

11. How do you move to the end of a sentence?

12. How do you move to the beginning of the file?

13. How do you move forward one view screen?

14. How do you leave EMACS?

Appendix E

Name _____

Date _____

How well do you know EMACS since you did this experiment?

EMACS Quiz (Form A)

1. How do you get into EMACS?

2. What's CTRL L? What does the L stand for?

3. What's ESC F? What does the F stand for?

4. What's ESC D? What does the D stand for?

5. What's CTRL P? What does the P stand for?

6. How do you insert new characters?

7. What's CTRL A? What does the A stand for?

8. What's ESC >? What does the > stand for?

9. What's CTRL F? What does the F stand for?

10. What's CTRL @ CTRL W? What do @ and W stand for?

11. What's ESC V? What does the V stand for?

12. What's ESC A? What does the A stand for?

Name _____

Date _____

How well do you know EMACS since you did this experiment?

EMACS Quiz (Form B)

1. What's CTRL E? What does the E stand for?

2. What's CTRL @, CTRL W, CTRL Y? What do @, W, and Y stand for?

3. What's CTRL XS? What does it stand for?

4. What's ESC D? What does the D stand for?

5. What's CTRL N? What does the N stand for?

6. What's CTRL G? What does the G stand for?

7. What's ESC B? What does the B stand for?

8. If you type CTRL C, what can you do to recover?

9. What's CTRL B? What does the B stand for?

10. What's CTRL K? What does the K stand for?

11. What's ESC E? What does the E stand for?

12. What's ESC <? What does the < stand for?

13. What's CTRL V? What does the V stand for?

14. What's CTRL XS CTRL XZ? What do they stand for?